NAME

perlsource - A guide to the Perl source tree

DESCRIPTION

This document describes the layout of the Perl source tree. If you're hacking on the Perl core, this will help you find what you're looking for.

FINDING YOUR WAY AROUND

The Perl source tree is big. Here's some of the thing you'll find in it:

C code

The C source code and header files mostly live in the root of the source tree. There are a few platform-specific directories which contain C code. In addition, some of the modules shipped with Perl include C or XS code.

See *perlinterp* for more details on the files that make up the Perl interpreter, as well as details on how it works.

Core modules

Modules shipped as part of the Perl core live in four subdirectories. Two of these directories contain modules that live in the core, and two contain modules that can also be released separately on CPAN. Modules which can be released on cpan are known as "dual-life" modules.

* *lib/*

  This directory contains pure-Perl modules which are only released as part of the core. This directory contains *all* of the modules and their tests, unlike other core modules.

* *ext/*

  This directory contains XS-using modules which are only released as part of the core. These modules generally have their *Makefile.PL* and are laid out more like a typical CPAN module.

* *dist/*

  This directory is for dual-life modules where the blead source is canonical. Note that some modules in this directory may not yet have been released separately on CPAN.

* *cpan/*

  This directory contains dual-life modules where the CPAN module is canonical. Do not patch these modules directly! Changes to these modules should be submitted to the maintainer of the CPAN module. Once those changes are applied and released, the new version of the module will be incorporated into the core.

For some dual-life modules, it has not yet been determined if the CPAN version or the blead source is canonical. Until that is done, those modules should be in *cpan/*.

Tests

The Perl core has an extensive test suite. If you add new tests (or new modules with tests), you may need to update the *t/TEST* file so that the tests are run.

* Module tests

  Tests for core modules in the *lib/* directory are right next to the module itself. For example, we have *lib/strict.pm* and *lib/strict.t*.

  Tests for modules in *ext/* and the dual-life modules are in *t/* subdirectories for each module, like a standard CPAN distribution.

* *t/base/*

  Tests for the absolute basic functionality of Perl. This includes `if`, basic file reads and writes,

simple regexes, etc. These are run first in the test suite and if any of them fail, something is *really* broken.

* *t/cmd/*

  Tests for basic control structures, `if/else`, `while`, subroutines, etc.

* *t/comp/*

  Tests for basic issues of how Perl parses and compiles itself.

* *t/io/*

  Tests for built-in IO functions, including command line arguments.

* *t/mro/*

  Tests for perl's method resolution order implementations (see *mro*).

* *t/op/*

  Tests for perl's built in functions that don't fit into any of the other directories.

* *t/opbasic/*

  Tests for perl's built in functions which, like those in *t/op/*, do not fit into any of the other directories, but which, in addition, cannot use *t/test.pl*,as that program depends on functionality which the test file itself is testing.

* *t/re/*

  Tests for regex related functions or behaviour. (These used to live in t/op).

* *t/run/*

  Tests for features of how perl actually runs, including exit codes and handling of PERL* environment variables.

* *t/uni/*

  Tests for the core support of Unicode.

* *t/win32/*

  Windows-specific tests.

* *t/porting/*

  Tests the state of the source tree for various common errors. For example, it tests that everyone who is listed in the git log has a corresponding entry in the *AUTHORS* file.

* *t/lib/*

  The old home for the module tests, you shouldn't put anything new in here. There are still some bits and pieces hanging around in here that need to be moved. Perhaps you could move them? Thanks!

* *t/x2p*

  A test suite for the s2p converter.

Documentation

All of the core documentation intended for end users lives in *pod/*. Individual modules in *lib/*, *ext/*, *dist/*, and *cpan/* usually have their own documentation, either in the *Module.pm* file or an accompanying *Module.pod* file.

Finally, documentation intended for core Perl developers lives in the *Porting/* directory.

Hacking tools and documentation

The *Porting* directory contains a grab bag of code and documentation intended to help porters work on Perl. Some of the highlights include:

* *check**

  These are scripts which will check the source things like ANSI C violations, POD encoding issues, etc.

* *Maintainers*, *Maintainers.pl*, and *Maintainers.pm*

  These files contain information on who maintains which modules. Run `perl Porting/Maintainers -M Module::Name` to find out more information about a dual-life module.

* *podtidy*

  Tidies a pod file. It's a good idea to run this on a pod file you've patched.

Build system

The Perl build system starts with the *Configure* script in the root directory.

Platform-specific pieces of the build system also live in platform-specific directories like *win32/*, *vms/*, etc.

The *Configure* script is ultimately responsible for generating a *Makefile*.

The build system that Perl uses is called metaconfig. This system is maintained separately from the Perl core.

The metaconfig system has its own git repository. Please see its README file in *http://perl5.git.perl.org/metaconfig.git/* for more details.

The *Cross* directory contains various files related to cross-compiling Perl. See *Cross/README* for more details.

AUTHORS

This file lists everyone who's contributed to Perl. If you submit a patch, you should add your name to this file as part of the patch.

MANIFEST

The *MANIFEST* file in the root of the source tree contains a list of every file in the Perl core, as well as a brief description of each file.

You can get an overview of all the files with this command:

```
% perl -lne 'print if /^[^\/]+\.[ch]\s+/' MANIFEST
```